Adaptive simplification of GPS trajectories with geographic context – a quadtree-based approach

Fu, Cheng ; Huang, Haosheng ; Weibel, Robert

Abstract: Big GPS trajectory datasets can have redundant spatio-temporal information for applications, which requires simplification as a key preprocessing for modeling. Many existing simplification methods focus on the geometric information from a trajectory per se. Conversely, methods considering geographic context often fail to provide spatially adaptive simplification, or require complex parameter settings to achieve this task. This study proposes a novel two-stage adaptive trajectory simplification method embedding spatial indexing, enrichment, and aggregation in an integrated process. The first stage employs a quadtree for the subdivision depending on the density of geographic context features (i.e. POIs), leading to a variable-resolution representation of the area. The second stage aggregates trajectory waypoints locating in the same quadtree leaf node into a representative point, making the aggregation adapting to the spatial layout of the geographic feature in the first stage. Evaluation with a real-world vehicle trajectory dataset shows that the proposed approach can automatically simplify trajectory segments at variable compression ratios with greater simplification in areas with sparse context features (e.g. rural) and less simplification in areas with dense context features (e.g. urban). More importantly, the method can still preserve inter-trajectory distances between original trajectories and simplified ones, while significantly reducing the computing time.

DOI: https://doi.org/10.1080/13658816.2020.1778003

Adaptive simplification of GPS trajectories with geographic context – a quadtree-based approach

Cheng Fu[a]*, Haosheng Huang[a,b], and Robert Weibel[a]

[a]*Department of Geography, University of Zurich, Zurich, Switzerland;* [b]*Department of Geography, Ghent University, Ghent, Belgium*

* cheng.fu@geo.uzh.ch

Adaptive simplification of GPS trajectories with geographic context – a quadtree-based approach

Big GPS trajectory datasets can have redundant spatio-temporal information for applications, which requires simplification as a key preprocessing for modeling. Many existing simplification methods focus on the geometric information from a trajectory per se. Conversely, methods considering geographic context often fail to provide spatially adaptive simplification, or require complex parameter settings to achieve this task. This study proposes a novel two-stage adaptive trajectory simplification method embedding spatial indexing, enrichment, and aggregation in an integrated process. The first stage employs a quadtree for the subdivision depending on the density of geographic context features (i.e., POIs), leading to a variable-resolution representation of the area. The second stage aggregates trajectory waypoints locating in the same quadtree leaf node into a representative point, making the aggregation adapting to the spatial layout of the geographic feature in the first stage. Evaluation with a real-world vehicle trajectory dataset shows that the proposed approach can automatically simplify trajectory segments at variable compression ratios with greater simplification in areas with sparse context features (e.g., rural) and less simplification in areas with dense context features (e.g., urban). More importantly, the method can still preserve inter-trajectory distances between original trajectories and simplified ones, while significantly reducing the computing time.

Keywords: trajectory simplification; adaptive simplification; quadtree; geographic context; points of interest (POIs)

1. Introduction

Recent years have witnessed a great increase in the number of GPS (Global Positioning System)-enabled devices in daily lives, which leads to an unprecedented scale of trajectory data about various moving objects, such as humans, vehicles, and animals. The increasing availability of big GPS trajectory datasets has boosted applications for human mobility pattern mining (Zheng *et al.* 2008, Guidotti *et al.* 2017), driving behavior analysis (Yuan *et al.* 2010), transportation volume modeling (Fan *et al.* 2019), transportation mode detection (Das and Winter 2016), animal movement ecology studies (Nathan *et al.* 2008, Demšar *et al.*

2015), etc. Techniques for revealing the spatio-temporal patterns of trajectories, such as trajectory clustering (Morris and Trivedi 2009, Pelekis *et al.* 2011, Toohey and Duckham 2015), often encounter significant engineering challenges when being adapted for analysis of big trajectory data. The computational complexity of trajectory distances often fits $O(n^2)$ (Besse *et al.* 2016), in contrast to the merely linear increase of computational capacity. Even for the popular distributed computing infrastructures for high-performance parallel computing, such as Hadoop and Spark (Zaharia *et al.* 2016), this computational complexity is still very significant. In addition, the inter-node data exchange at a shuffling stage in both Hadoop and Spark jobs is very time-consuming (Sun *et al.* 2016). One intuitive but common strategy is to simplify a trajectory by reducing its waypoints while preserving its spatial-temporal characteristics.

Initially, trajectory simplification algorithms adopted common algorithms from cartographic line simplification, such as the classic Douglas-Peucker (DP) algorithm (Douglas and Peucker 1973) and its descendants, leading to algorithms such as the top-down time-ratio algorithm (TD-TR, Meratnia & de By, 2004) and the direction-preserving trajectory simplification (DPTS, Long *et al.* 2013). However, these simplification algorithms only rely on the geometric information extracted from a trajectory per se, and fail to consider the geographic context where the movement happened. This might be undesirable for many applications where trajectories are only meaningful when their geographic context is considered, e.g., for human mobility analysis and transportation studies. Meanwhile, relating trajectories to their geographic context at an early stage will facilitate the analysis of their spatio-temporal patterns afterward.

Semantic enrichment by geographic context provides a promising way to trajectory simplification, and it tries to leverage the geographic context of a trajectory to facilitate trajectory compression (Alvares *et al.* 2007, Yuan *et al.* 2010, Zhang *et al.* 2018). A common

method in this respect is map matching, which snaps waypoints to the road network, and thus converts the discrete waypoint sequence of the raw trajectory to a road segment sequence (Richter *et al.* 2012, Quddus and Washington 2015, Sandu Popa *et al.* 2015). Besides road networks, other geographic data sources might also be employed, such as Points of Interest (POIs). Typically, the waypoints of a raw trajectory are first annotated with nearby POIs. After that, waypoints annotated with the same POIs are aggregated into a 'representative point' (i.e., a placeholder point), which leads to a simplified trajectory. The challenge here is mainly on identifying which POIs to use, considering that there might be a huge number of nearby POIs. A common strategy is either to apply a simple but universal rule such as joining to the nearest POI (Shang *et al.* 2015) or to require complex parameter settings such as taking the temporal information and the function of the POI into account for selecting the best match (Furletti *et al.* 2013).

This paper presents a novel semantic enrichment-based trajectory simplification method, which can simplify segments of a trajectory at different compression ratios according to their underlying geographic context. For example, with this method, a trajectory can be less simplified in certain geographic areas such as downtown, but more simplified in other areas, such as suburban and countryside. To achieve this type of adaptive simplification, the method uses a quadtree that recursively divides a geographic area of interest into four equal-size, rectangular cells, or quadrants, based on the number of POIs within the area and its cells, leading to a variable resolution representation of the area. The tessellation is regular and spatially exclusive (i.e., non-overlapping) at a given level. In addition, its subdivisions are easy to understand and visualize, compared to other common spatial indexing structures such as R-tree and k-d tree (Samet 2006). Therefore, the method does not require complex parameter settings but *practically* only needs one parameter to control the minimum granularity of tessellation by choosing the maximal POI points each

leaf can contain when simplifying trajectories. It is particularly suitable for simplifying trajectories with large geographic coverage, e.g., a trajectory of a truck that travels around a whole country. This study also investigates to what degree this new method may preserve the intra- and inter-trajectory distances with different distance metrics.

The remainder of the paper is structured as follows: Section 2 summarizes related work. Section 3 introduces the proposed methodology. The evaluation and its results are presented in Section 4 and discussed in Section 5. Section 6 summarizes the main findings and presents ideas for future work.

2. Related work

2.1 Trajectory modeling

Although a raw GPS trajectory is presented as a chronological order of geographic locations recorded by a GPS logger, a trajectory is commonly modeled from three perspectives: geometry, movement parameters, and semantics. From the geometry perspective, a trajectory is modeled as a series of spatio-temporal points denoted as (l_i, t_i), where l_i is a tuple of coordinates in 2D or 3D space, e.g., (x, y), (latitude, longitude), or (latitude, longitude, altitude), while t_i is a timestamp. The raw GPS trajectory records thus are instances of the geometry model.

From the movement parameters perspective, a trajectory can be modeled as a time series of different movement-related parameters, such as velocity, speed, and acceleration, which can be further coded by symbolic presentations (Dodge *et al.* 2009), denoted as (a_i, t_i), where a is a movement parameter, and a_i is one of the symbolic codes of the movement parameter. The movement parameter model thus represents the attributes of trajectories but excludes the information on locations. Data for such a model can be derived from the

geometry model by calculating the movement parameters following the method described in Dodge *et al.* (2009).

Finally, from the semantic perspective, a trajectory is modeled as a series of geographic entities ordered along time, denoted as (p_i, t_i), where p_i is a generalized geographic entity with the sense of *place* (Tuan 1975) that may provide different affordances (Gibson 1979) and can be represented by proxies such as POIs, areas of interest (AOIs), entities recognized by personal experience, landmarks in a landscape, etc. The semantic model generalizes locations in space into geographic entities and thus transforms a trajectory from a sequence of locations to a sequence of places, travel/behavioral modes, and/or activities (Siła-Nowicka *et al.* 2016). To process such a model, additional geographic data sources have to be introduced into data processing, such as POI data, land use data, and administrative boundaries.

2.2 Trajectory simplification algorithms

Trajectory simplification can be categorized into offline mode and online mode. The offline mode is for simplifying archived trajectories as a whole, while the online mode is for processing streaming waypoints. This section only focuses on existing offline-mode trajectory simplification methods.

One of the most popular trajectory simplification methods is the Douglas-Peucker (DP) algorithm. The algorithm uses a geometrical metric perpendicular Euclidean distance (PED) and guarantees the upper bound error of the waypoints. Essentially, DP is a cartographic line simplification algorithm preserving the geometric shape of the line but ignoring the time dimension for the trajectory. The top-down time-ratio algorithm (TD-TR) extends DP by using synchronized Euclidean distance (SED) to replace PED as the error metric in DP so that temporal information is also modeled and preserved. Besides these two classic algorithms, there are other algorithms aiming to speed up the compression process by

reducing the time complexity utilizing the properties of the convex hulls in the trajectory for DP (DP-Hull, Hershberger & Snoeyink 1992) and utilizing an approximation of SED for TD-TR (Chen *et al.* 2012). In addition to the geometric feature preservation and geometric-temporal feature preservation, there are also algorithms aiming at preserving the moving direction of the trajectory, such as the direction-preserving trajectory simplification (DPTS).

Empirical comparisons using real-life data sets show that there is no rule-of-thumb algorithm for preserving all trajectory features that basic DP and TD-TR can still perform well on preserving PED, SED, direction, and the speed profile compared to many other more recent algorithms; and the performance may also vary depending on different data sets (Zhang *et al.* 2018). This explains the demand for more trajectory simplification algorithms preserving different features of the original trajectory, while in the meantime the basic algorithms are still used as baselines.

2.3 Semantic enrichment of trajectories

Semantic enrichment provides another way of simplifying querying, analyzing, and mining trajectories by introducing geographic sources and spatial semantics into trajectory modeling (Alvares *et al.* 2007, Purves *et al.* 2014, Dodge 2019). Semantic enrichment annotates trajectories with the special semantics of geographic entities by spatio-temporal co-location.

Map matching is a typical trajectory simplification method, which defines constraints on possible movement of the moving object, e.g., movement of a vehicle needs to follow the road network. The number of intermediate points can therefore be reduced to the number of road intersections the trajectory has crossed. Map matching, however, requires a complete representation of the road network.

Besides road networks as the main geographic source for the map matching task, other geographic sources can either be intrinsically inferred from the spatio-temporal patterns of the waypoints per se, or imported from external data sources for other general semantic

enrichment tasks. For the first method, the patterns may be dense waypoint clusters that can be further inferred as home, workplace, or other places (Andrienko *et al.* 2007, Zhuang *et al.* 2017). Further constraints can be added to the waypoint clusters, such as sharing similar movement properties (Vrotsou *et al.* 2015).

The second common method involves external geographic data to append external attributes to the waypoints based on the spatial relationship between the waypoints and the subdivision areal units. One common technique is to annotate the waypoints or segments of a trajectory with nearby POIs (Furletti *et al.* 2013, Krueger *et al.* 2015). Semantic enrichment thus can also be used as an approach for adaptive trajectory simplification by aggregating waypoints with the same semantic annotation into a representative point. For example, Rothermel *et al.* (2012) employed a trajectory simplification strategy adapting to the abnormality of a moving object's current position and previous position as the semantic context. Lin *et al.* (2016) partition a trajectory based on the speed profile before applying the Douglas-Peucker algorithm to each partition separately. However, these methods are not adaptive to the spatial context associated with the location of the moving objects.

2.4 Research gaps

Trajectory simplification is subject to the application of trajectory analysis to determine which features should be better preserved and to what degree the number of waypoints should be reduced. Some applications, such as truck monitoring and visualization, require an adjustable simplification strategy where a trajectory is less simplified in certain geographic areas such as downtown but more simplified in other areas, such as in suburban areas and the countryside. For analyzing mobility patterns of a big trajectory data set that covers a whole country or multiple cities rather than only a single city as the commonly used Geolife data set (Zheng *et al.* 2008), more geographically connotated context such as the urban-rural dichotomy should be considered.

Summarizing existing research on trajectory simplification, we find that: 1) Many existing trajectory simplification methods focus on the geometric information extracted from a trajectory *per se*, without involving the geographic context of the moving object. And 2) methods considering geographic context often fail to provide adaptive simplification of a trajectory, or require complex parameter settings to achieve this task.

This article aims to address these research gaps, and proposes a quadtree-based method to automatically simplify segments of a trajectory at different compression ratios, in accordance with their underlying geographic context.

3. Adaptive trajectory simplification

Geographic phenomena often unevenly distribute over space, leading to spatial heterogeneity. When analyzing movement behaviors, we also tend to pay different attention to different parts of a trajectory. For example, in traffic monitoring applications, vehicle positions are paid more attention while being in places such as central business districts (CBDs), but are given less consideration outside urban places such as on interstate highways. From the positional uncertainty perspective, given a particular positioning accuracy, vehicles in urban areas have more uncertainty in terms of which particular road they are located on or which particular building they are nearby, but have less associated uncertainty when on highways outside the city. Therefore, this work uses a variable-resolution subdivision adopting the practice used by Soleymani *et al.* (2014) and the theory summarized by Jiang and Brandt (2016) and Li *et al.* (2018). Instead of the manually defined subdivision in Soleymani *et al.* (2014), our variable-resolution subdivision is generated by an automated process. The approach is based on the principle of subsampling an originally fine-grained trajectory representing a micro-scale rendition of movement to obtain more coarse-grained representations at the meso and finally the macro-scale.

We further argue that such variable resolution spatial subdivisions are subject to the research context. For instance, in a scenario of monitoring a truck trajectory in a metropolitan region, small grids can be used for aggregating geographic features within the city centers while larger grids are used for modeling suburban and countryside areas (Figure 1.A).

Additionally, by changing the minimum resolution of a subdivision, cross-scale effects can also be explored. For example, all grids in Figure 1.A at the current scale may be further split into smaller grids to generate a new subdivision with overall finer resolution, as shown in Figure 1.B. The subdivision in Figure 1.A may be referred to as having a meso-scale resolution, while the subdivision of Figure 1.B has a micro-scale resolution. A meso-scale simplification leads to a higher compression ratio than a micro-scale simplification, as waypoints are aggregated to a coarser sampling subdivision.

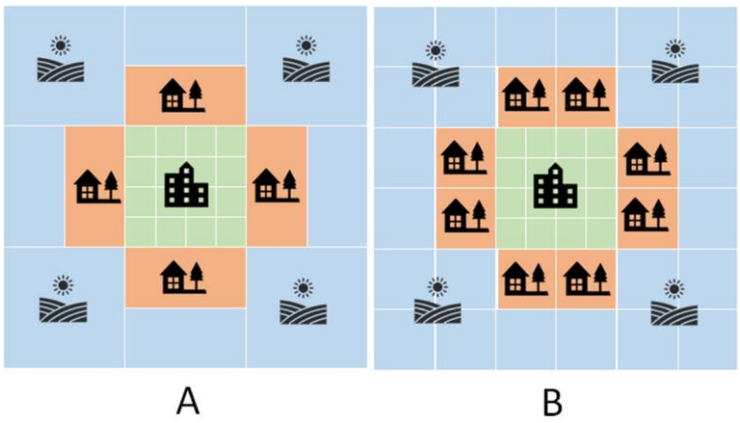


Figure 1. Schematic variable-resolution subdivisions of the simplified urban-rural dichotomy to fit the demand of different modeling purposes. A) A variable-resolution spatial subdivision model, e.g., for a scenario of modeling truck movement covering a large geographic area. B) The same scenario of modeling truck movement as in A), but with a finer scale (finer minimum resolution). Green grid cells denote the downtown area. Orange grid cells correspond to suburban areas. The blue grids correspond to the countryside.

3.1 Overview of the methodology

Figure 2 shows the methodology of the proposed adaptive trajectory simplification method. The proposed method takes a raw GPS trajectory and external geographic information as inputs, and returns a simplified trajectory. The method has two stages: quadtree-based subdivision of the geographic area of interest (Section 3.2), and semantic aggregation of raw trajectory waypoints (Section 3.3). For the first stage, this paper particularly uses POI data in the area of interest (i.e., Europe in the empirical evaluation) as a source for the quadtree-based subdivision to model the urban-rural dichotomy. Note, however, that alternative data may also be used to model the spatial heterogeneity of other geographic phenomena. This stage needs to be carried out only once. In the second stage (Section 3.3), for each raw trajectory to be simplified, its waypoints are associated with the quadtree leaf (i.e., a cell) they are located in. Neighboring waypoints associated with the same cell are then aggregated into a representative point, resulting in a simplified trajectory. By changing the quadtree splitting threshold in the first stage, a finer or coarser subdivision of the geographic area can be achieved, which can be used to control the overall compression ratio of the simplification results.

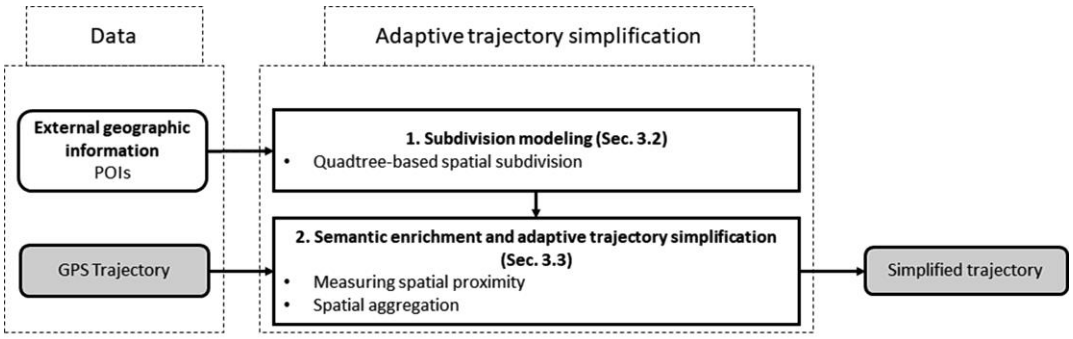


Figure 2. Overview of the methodology

3.2 Quadtree-based subdivision with POIs

The spatial heterogeneity of POIs reflects the spatial layout of urban and rural areas (Long *et al.* 2016). The density of POIs in cities also exhibits the socio-economic function of different urban areas, e.g., whether a region is a CBD, local commercial center, or residential areas. In the following, we employ a quadtree to model the spatial heterogeneity of POI distribution within a geographic area of interest, which leads to a variable-resolution representation of the geographic area.

A quadtree is a tree data structure in which each node either has exactly four children, or has no children (i.e., a leaf node) (Finkel and Bentley 1974). Quadtrees are often used for spatial indexing. They partition a two-dimensional space by recursively subdividing it into four quadrants, or cells. The partitioning continues if the number of contained data points in a cell exceeds a pre-defined splitting threshold. Once the partitioning finally stops, the leaf cells together form a subdivision of the space.

Applying the concept of quadtrees on POI data helps to subdivide a geographic area into non-overlapping cells with variable sizes, reflecting the spatial heterogeneity of the POIs within the area. Conceptually, this quadtree-based subdivision with POIs (in short, POI-quadtree) can be used as a proxy to model the urban-rural dichotomy of a geographic area: As can be seen in Figure 3, regions with higher density of POIs are divided into small leaf cells and usually denote an urban area, while regions with lower density of POIs are partitioned into larger cells and usually represent country-side or sub-urban areas. This matches the layout of the conceptual model in Figure 2. In the evaluation (Section 4.2), we quantitatively investigated how well the POI-quadtree subdivision reflects the urban-rural dichotomy of a geographic area, using the official CORINE Land Cover 2018 dataset (Copernicus 2019). This variable-resolution representation of the geographic area allows to adaptively simplify segments of a trajectory at different compression ratios (Section 4.3).

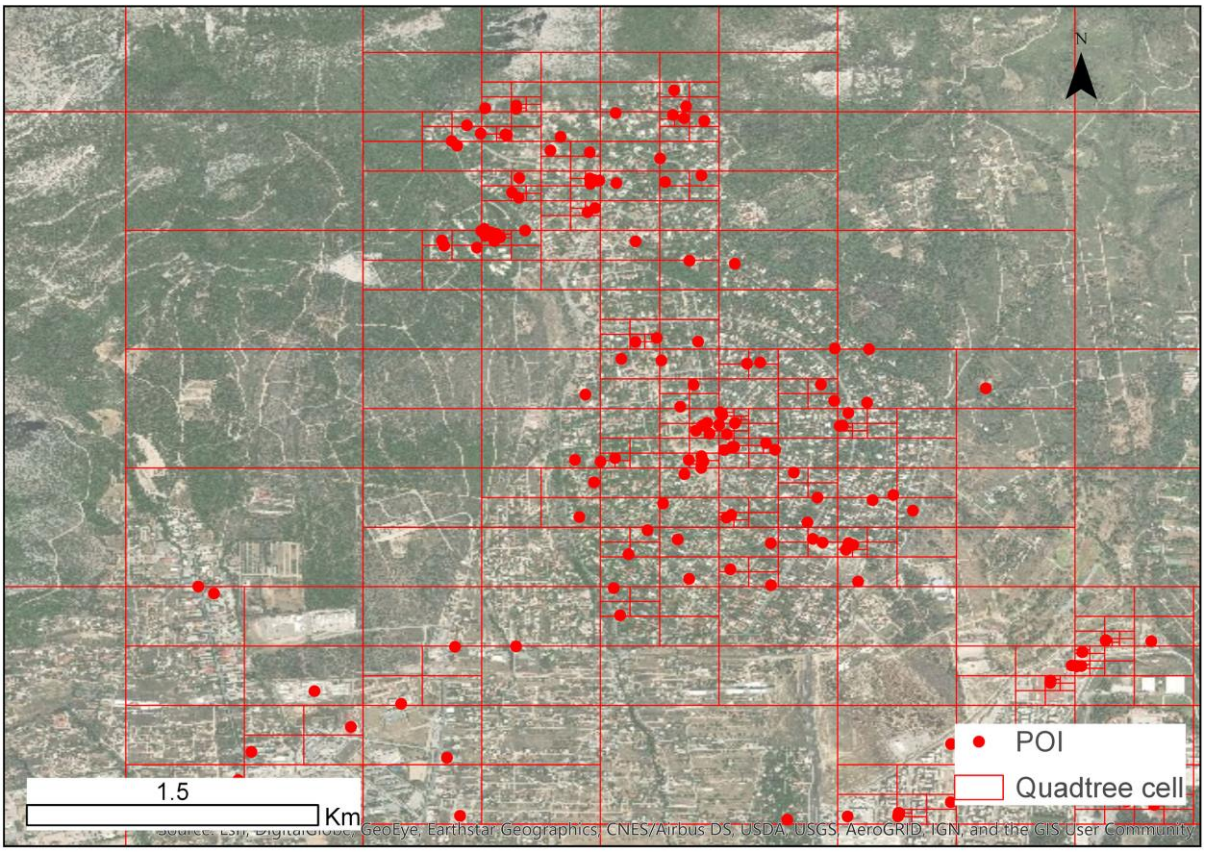


Figure 3 An example of POIs from OpenStreetMap (OSM) that are labeled as "POI" in Geofabrik (Ramm 2017)taxonomy and the corresponding POI-quadtree. The urban area in the center is the Town of Thrakomakedones, located north of Athens, Greece. The POI-quadtree is a proxy to model the urban-rural dichotomy of a geographic area: Urban regions, which have a higher density of POIs, are divided into smaller leaf cells; while rural areas, which contain less POIs, are partitioned into bigger cells.

Appendix A shows the pseudo-code of this process. Two adjustments are made from the original quadtree building algorithm. Firstly, the spatial extent of the root is set as the extent of the world. A potential benefit of this is that two quadtrees built from two geographically exclusive (i.e., non-overlapping) POI sets can be merged together easily, e.g., merging the European POI-quadtree with the Asian POI-quadtree. Additionally, the built quadtree can be easily re-used for other research purposes (e.g., simplifying trajectories in other geographic areas), and allows for comparisons over different areas due to the fact that a

common reference frame (i.e., the whole world) is employed. Secondly, maximal depth of the quadtree is introduced to control the minimal size of the leaf cells. Usually, the minimal size of a leaf cell should not be smaller than the size of the moving object, or the accuracy limit of the tracking data used.

Taking a set of POIs and two parameters (i.e., splitting threshold ST, and maximal depth MD) as inputs, the algorithm recursively checks whether the number of contained POI data points exceeds the splitting threshold ST and whether it has not reached the maximal depth MD. ST defines the maximum capacity of data points (i.e., POIs in this paper) each leaf cell can hold. MD defines the maximal possible depth of the tree. If yes, the associated geographic area is partitioned into four quadrants, or cells by creating four child nodes, and the contained POI points are distributed to the corresponding child nodes. The output of the algorithm is the built POI-quadtree.

One important parameter of the POI-quadtree building method is the splitting threshold ST. By using different splitting thresholds, the POI-quadtree can provide different spatial resolutions. A larger threshold for POI-quadtree building corresponds to bigger leaf cells in the final POI-quadtree. In other words, a larger threshold leads to a coarser subdivision of the geographic area, while a smaller one results in a finer subdivision of the area. Figure 4 provides an example of this effect. A larger splitting threshold thus also leads to less tree depth and fewer tree nodes, which consumes less memory and takes less time for querying the quadtree. More importantly, different splitting thresholds lead to different overall compression ratios of a trajectory. On the one hand, this will give users some freedom to control the overall compression ratio of trajectories. On the other hand, users should carefully select a suitable splitting threshold for the POI-quadtree building, as the subdivision will also influence the spatial approximation of the original trajectory geometry.

Figure 4 Examples of the built quadtree with different splitting thresholds, using real POI data. A) splitting threshold 1 (denoted as QT-1); B) splitting threshold 10 (denoted as QT-10). The geographic area of the examples is located in the north part of Athens, Greece.

In practice, selecting a proper set of POIs is important for modeling different activities; more discussion of this point follows in Section 5. Some heuristics might be employed to choose a suitable splitting threshold. For example, when applying the POI-quadtree to simplify GPS trajectories, one needs to make sure that the smallest cell size should be neither smaller than the size of the moving object (e.g., a vehicle in this study), nor smaller than the GPS accuracy.

3.3 Semantic enrichment and aggregation

Once the POI-quadtree is built, the geographic area is subdivided into cells with variable sizes. In the following, we use this variable-resolution representation of the geographic area to simplify GPS trajectories. The idea is to aggregate neighboring waypoints that are located in the same cell into a 'representative' waypoint, and thus reduce the number of waypoints and simplify a trajectory.

Once space has been subdivided by the POI-quadtree, the subdivisions have an implicit spatial semantic embedding in the cell size and the spatial relationship with the POIs. Such semantics is not as explicit as the POI types and needs some interpretation. For the case

of Figure 5, the cells are associated with the two POIs. The green cell can be referred to as "the cell north of the town of Neo Pontos" in a description. With the conventional nearest neighbor method, the waypoints will always be associated with the two restaurants POIs on the right, even if the shortest distance between any of the waypoints and the POIs is over 2 km. However, if the waypoints are annotated by the quadtree cells, the semantics is closer to the concept that the waypoints are passing by the mountain area in the countryside rather than passing by a distant restaurant. After the waypoints are partitioned by the quadtree cell annotations, the waypoints sharing the same cell annotation can be aggregated into representative points by different methods for the purpose of simplification. For example, all waypoints in the green cell may be represented as one waypoint using the geometric centroid of the waypoints, the midpoint of the entry and the exit point, or simply the midpoint of the quadtree cell, depending on the purpose of the application.

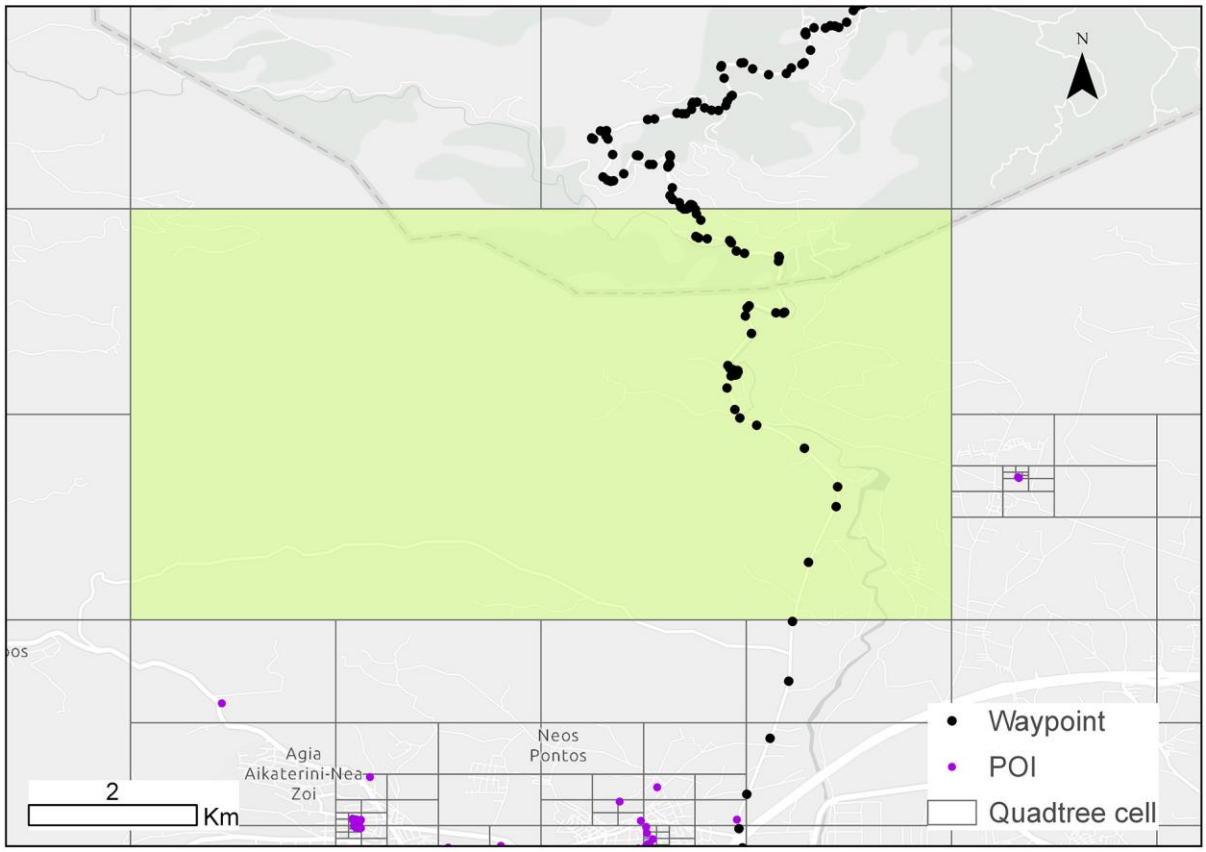


Figure 5 An example of the POI-quadtree leaf cells as the spatial semantics of trajectory waypoints near Neo Pontos, a suburban area in the north of the Elefsina city, Greece. See text for explanation.

Appendix B shows the pseudo-code of the algorithm. It takes a raw trajectory and the root of the built quadtree (after applying Algorithm 1, Appendix A) as inputs, and returns a simplified trajectory. No further parameter calibration is needed. The algorithm iteratively checks whether a waypoint is located in the same quadtree leaf node with its previous sibling waypoints, and aggregates them into a representative waypoint if a new leaf node appears. The representative waypoint might be a new point, taking the geometric centroid of the original waypoints as its location and the midpoint of time of the first and last original waypoints in the leaf node as its timestamp. Alternatively, it may simply be the middle/median point of the original waypoints. Meanwhile, the movement parameters such as speed can also be aggregated based on projected coordinates and attached as attributes for this new representative waypoint.

Note that to improve the association of the original waypoints to the leaf node in the quadtree, the nodes of the quadtree can be coded by the Morton code (i.e., z-order curve, Morton 1966), which will significantly improve the performance of the algorithm. Additionally, time constraints may also be added to prevent waypoints from being aggregated together if the time span between them exceeds some threshold.

4. Evaluation and results

We evaluated the proposed adaptive trajectory simplification method with real-world vehicle trajectories, addressing the following questions:

1) How well does the POI-quadtree subdivision scheme reflect the urban-rural dichotomy of a geographic area? (Section 4.2)

2) How does the POI-quadtree perform in terms of compression ratio, simplification errors, and preservation of waypoints in urban areas, compared to other trajectory simplification methods, such as DP and TD-TR? (Section 4.3)
3) How well can the proposed method preserve pairwise trajectory similarity while reducing the computational time? (Section 4.4)

We introduce the data and experimental setup in Section 4.1, and summarize the main results in Section 4.5.

4.1 Data and experimental setup

We used a real-world vehicle trajectory dataset, provided by Vodafone Innovus (https://www.vodafoneinnovus.com/), our partner in the Track & Know project (https://trackandknowproject.eu/), who offer fleet management services. The data are anonymous in accordance with the General Data Protection Regulation. Specifically, we randomly selected 500 trajectories from this dataset, whose spatial extent covers the whole of Greece (Figure 6). The mean trip length regarding the number of waypoints is 263 (SD = 215) within a range between 51 and 1889. The mean sampling rate is 24 seconds per waypoint (SD = 22 sec).

The POIs of Europe were extracted from OpenStreetMap (OSM) in January 2019, in order to honor the generality and the capacity to extend for the whole trajectory dataset in future work. A taxonomy of OSM point features from Geofabrik was used for filtering certain POI types, within which there are five top types: *places*, *POI*, *places-of-worship*, *nature*, *traffic-transport-and-power*. Only the type *POI* was selected for this study, resulting in 4,641,857 POIs, which include most public and commercial uses, such as public buildings, schools, shops, restaurants, hospitals, etc.

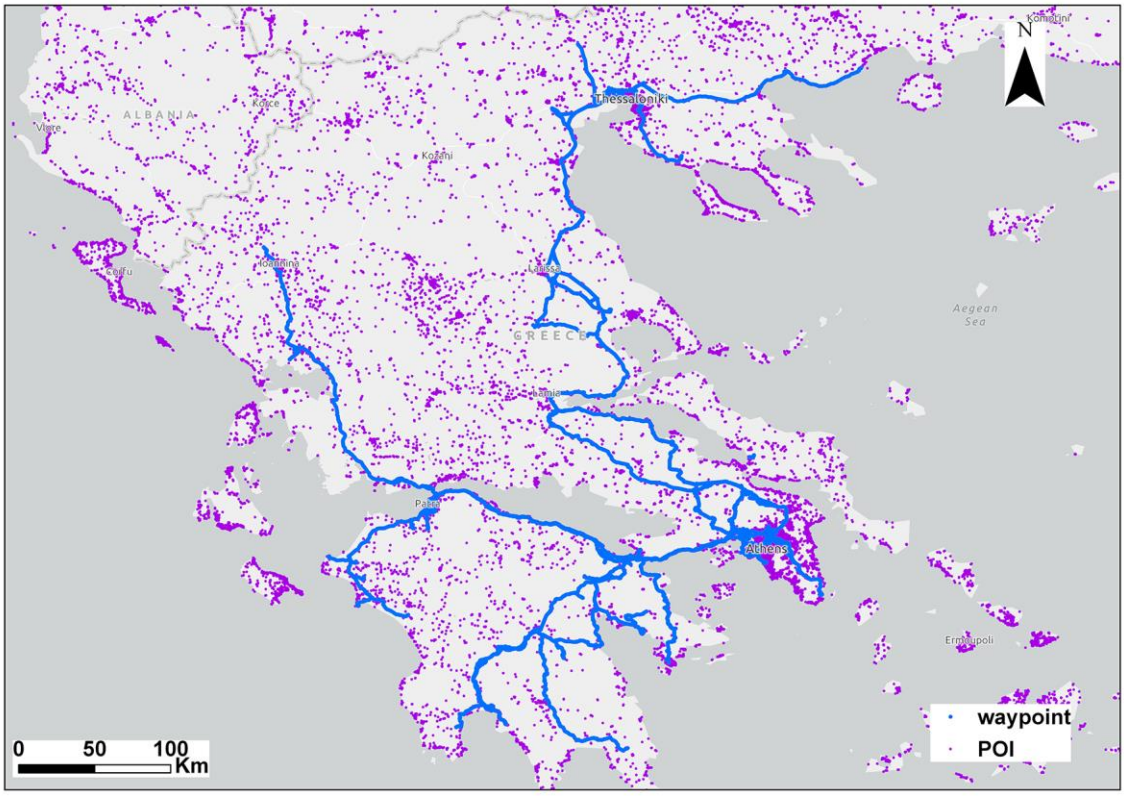


Figure 6 Waypoints of the selected trip sample and OSM POIs in Greece.

As a pilot study, the experiments were accomplished on a server with a 2.4 GHz 8-core CPU, 32 GB memory, and Ubuntu 16.04.5 LTS. The algorithms (i.e., the proposed POI-quadtree method and the two baseline methods DP and TD-TR) were implemented in Python 3.6.5.

For the POI-quadtree generation, the splitting threshold was set as 1, 5, 10, and 20 POIs for the sensitivity analysis, with the resulting subdivisions named QT-1, QT-5, QT-10, and QT-20, respectively. For the comparison with the DP algorithm, the maximum distance error threshold was set as 10, 50, 100, and 200 meters, named DP-10, DP-50, DP-100, and DP-200, respectively. Finally, for the TD-TR method, the maximum distance error threshold was set as 10, 50, 100, and 200 meters as well, and the maximum speed error threshold was set as 1, 5, 10, and 20 m/s, resulting in 16 combinations of parameters for TD-TR.

4.2 POI-quadtree generation

A complete POI-quadtree was built based on setting the splitting threshold as 1 POI, while constraining the maximum depth of the tree to 24 levels. The maximum depth constrains the minimum quadtree cell size to about 3 meters, which has a spatial scale similar to the width of a truck. This leads to a subdivision of the geographic area into cells with variable sizes, reflecting the spatial heterogeneity of POIs.

To evaluate how well this subdivision reflects the urban-rural dichotomy of the geographic area, we spatially joined each quadtree cell with the CORINE Land Cover 2018 dataset (Copernicus 2019)to compute the areal percentage of urban areas for each cell (i.e., the "1. Artificial Surfaces" class in CORINE, excluding a subclass "1.2.2 Road and rail networks and associated land"). CORINE land cover is an official dataset produced by the European Environment Agency (EEA). As expected, the spatial layout of the built quadtree cells corresponds to the spatial heterogeneity of the POIs, manifesting that the urban areas have more small cells, while the suburban and rural areas have more large cells. Taking the suburban area of Athens, Greece, as an example (Figure 7.A), it can be observed that there is a spatial overlap of high percentage urban area and high-density POI area. As a finer spatial resolution example at the edge of the city and its suburbs shows (Figure 7.B), quadtree cells closer to the city center (towards the south) is smaller than the counterparts in the north.

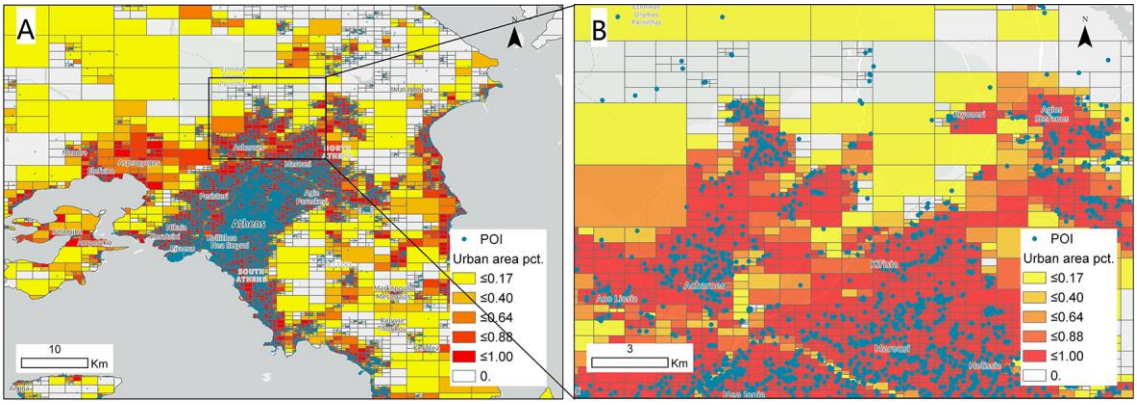


Figure 7 A visualization of the built POI-quadtree (QT-1) around Athens, Greece. Quadtree cells are spatially joined with the CORINE Land Cover 2018 dataset to extract the areal percentage of urban areas for each cell. A) POIs and the quadtree cells around Athens, Greece. B) POIs and the quadtree cell of a sample region at the edge of the urban and suburban area of Athens, Greece.

Quantitative observation of the areal percentage of urban land use for quadtree leaf cells at different levels also confirms that the areal percentage of urban land per cell grows along with increasing depth. After depth level 19, the mean proportion of urban land takes up more than 80% of a quadtree cell, though the mean slightly decreases after depth level 22, which might occur due to the modifiable areal unit problem (MAUP, Wong 2009) caused by cells in the suburban area (Figure 8). Nevertheless, the quantitative result further confirms our argument that the POI-quadtree is a good proxy for the layout of the urban-rural context, whereby small quadtree leaf cells correspond to the urban area.

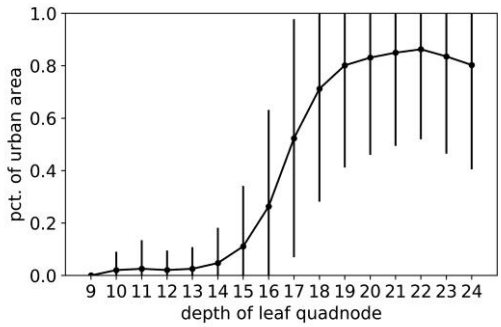


Figure 8 Average areal percentage of urban land use for leaf cells of the built POI-quadtree at different depths with a one-standard-deviation error bar. Note that the depth of a leaf cell/quadnode reflects the size of the cell, with deeper levels corresponding to smaller cells.

4.3 Comparing POI-quadtree simplification with DP and TD-TR

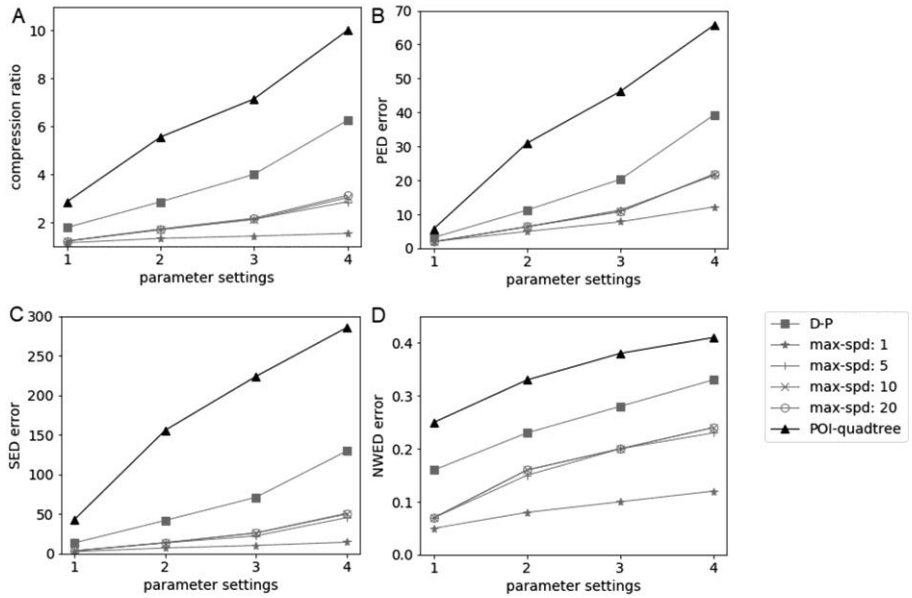
This section reports on the comparison of the proposed POI-quadtree method with the DP and TD-TR algorithms. As shown in a comprehensive comparison in Zhang *et al.* (2018), the basic DP and TD-TR still perform very well on preserving PED, SED, direction and speed profiles compared to many other more recent trajectory simplification algorithms. They can thus serve as a baseline for our evaluation.

4.3.1 Compression ratio and simplification error

When comparing these algorithms, we mainly focus on their compression ratio, as well as the error between a raw trajectory with its simplified result. The compression ratio is computed as the ratio of the number of waypoints of the raw trajectory and the number of waypoints of the simplified one. For measuring the simplification error, we used the PED and SED metrics suggested by Zhang *et al.* (2018) and Leichsenring and Baldo (2019). The PED metric measures the average shortest Euclidean distance from each original waypoint to its simplified segment. As an extension of PED, SED further takes the time cost of the movement into account, acknowledging that both the waypoint and the simplified segment are embedded in a spatial-temporal context. Meanwhile, we also used the NWED

(normalized weighted edit distance) metric proposed by Dodge *et al.* (2012) to see how well the movement parameter profiles (i.e., speed in this study) of a raw trajectory are preserved in the simplified version.

Figure 9 shows the comparison results. In general, the compression ratio of POI-quadtree simplification with a splitting threshold of 1 is between the results of DP-50 and DP-100, which is also similar to the TD-TR with a 200-meter maximum distance error and 5, 10, and 20 m/s maximum speed error (Figure 9.A). Accordingly, the POI-quadtree with splitting threshold 1 has similar PED, SED, and NWED errors as the DP-50 and DP-100 (Figure 9.B-D). The overall performance of the POI-quadtree with splitting threshold 5 is similar to DP-200.



Method/Parameter settings	1	2	3	4
D-P	10 m	50 m	100 m	200 m
max-spd: 1,5,10,20	10 m	50 m	100 m	200 m
POI-quadtree	1	5	10	20

Figure 9 A: The median compression ratio of all trajectories with three simplification algorithms (DP, TD-TR and POI-quadtree). B: median of the mean PED of all trajectories. C: median of the mean SED of all trajectories. D: median NWED of the mean NWED of all trajectories. The same parameter setting number corresponds to different parameter settings for different algorithms, as referenced in the table.

4.3.2 Adaptive simplification based on urban-rural geographic context

Figure 10 provides two examples, comparing the simplification effect of DP-200, TD-TR-D200-S5 and POI-quadtree simplification with splitting threshold 5. DP-200 and QT-5 have similar performance in terms of compression ratio and simplification errors (PED, SED, and NWED), while TD-TR-D200-S5 achieves considerably lower compression, as seen in Figure 9. However, the POI-quadtree, as designed, resulted in more waypoints being retained in urban areas (the segments near the city of Elefsina) and fewer waypoints retained outside towns, such as in the mountainous area in the upper part of Figure 10.A.4. In contrast, both DP-200 and TD-TR-D200-S5 maintain the curvy trajectory along the mountain road well but have less detail near the city of Elefsina. In addition, the TD-TR method even retained more waypoints in the straight road between the city and the mountains. In the example of another urban area near Nea Ionia in the northern part of Athens (Figure 10.B), there is a cluster of raw waypoints near the top-center of the map suggesting several potential events such as traffic congestion. The POI-quadtree has more representative points remaining to indicate the details of the actual path, while DP-200 only retained one. TD-TR retained the number of waypoints somewhere between the other two methods.

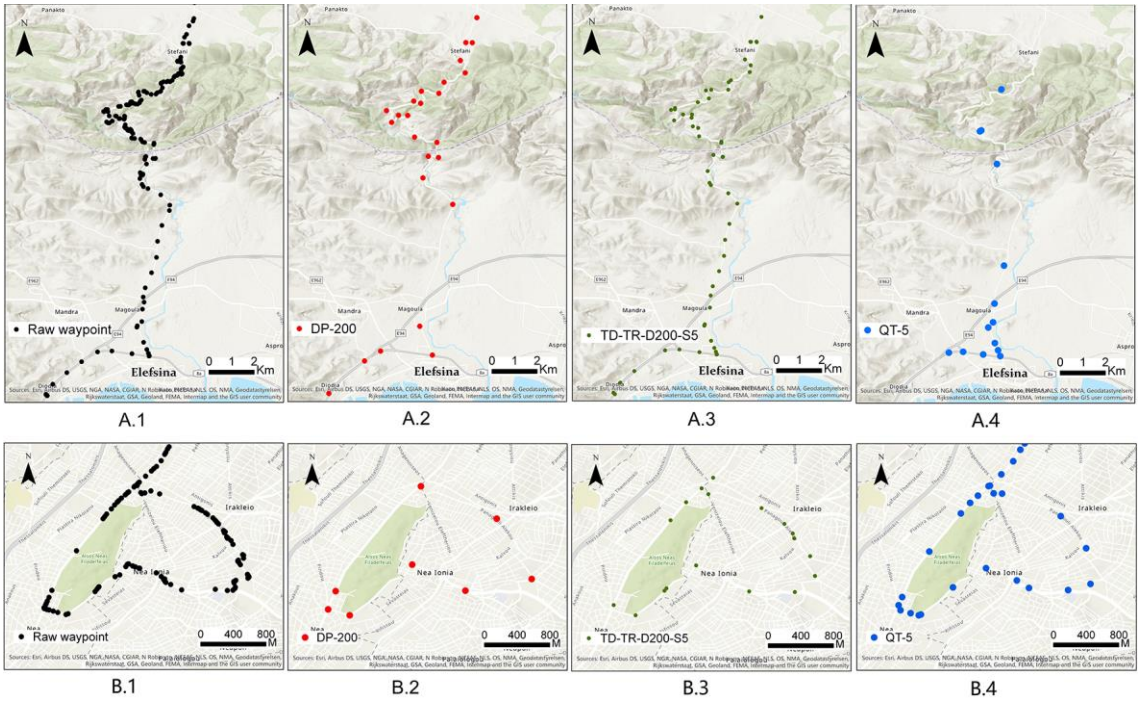


Figure 10 Two visual comparisons of simplified trajectories using DP-200, TD-TR with a 200-meter maximum distance error and 5-m/s maximum speed error (TD-TR-D200-S5), and a POI-quadtree with splitting threshold 5 (QT-5). A) The region near the city of Elefsina; B) Nea Ionia in the northern part of Athens, Greece.

An in-depth quantitative comparison of the simplification results shows that the simplified trajectories by DP-200 lead to a slightly lower percentage (2% change; not found to be significant by a one-tailed t-test with $p=0.27$) of waypoints located in the urban land use parcels as marked in CORINE Land Cover 2018 (denoted as *urban waypoint* in Figure 11.A). For the POI-quadtree, the percentage of waypoints located in the urban area increases from 35% to 42% (7% change; significant with $p<0.001$ using a one-tailed t-test) after simplification.

At the level of individual waypoints, Figure 11.B shows that the increase mainly happens on the trajectories that initially have a low percentage of urban waypoints. For example, the trajectories with 20% raw urban waypoints experienced an increase to 30% for most cases after being simplified by the POI-quadtree method. Compared to DP, which can

still be considered as a valid baseline trajectory simplification method, the proposed POI-quadtree algorithm retained more waypoints in urban areas and fewer waypoints in rural areas. In other words, it can automatically simplify segments of a trajectory at different compression ratios according to their underlying urban-rural geographic contexts.

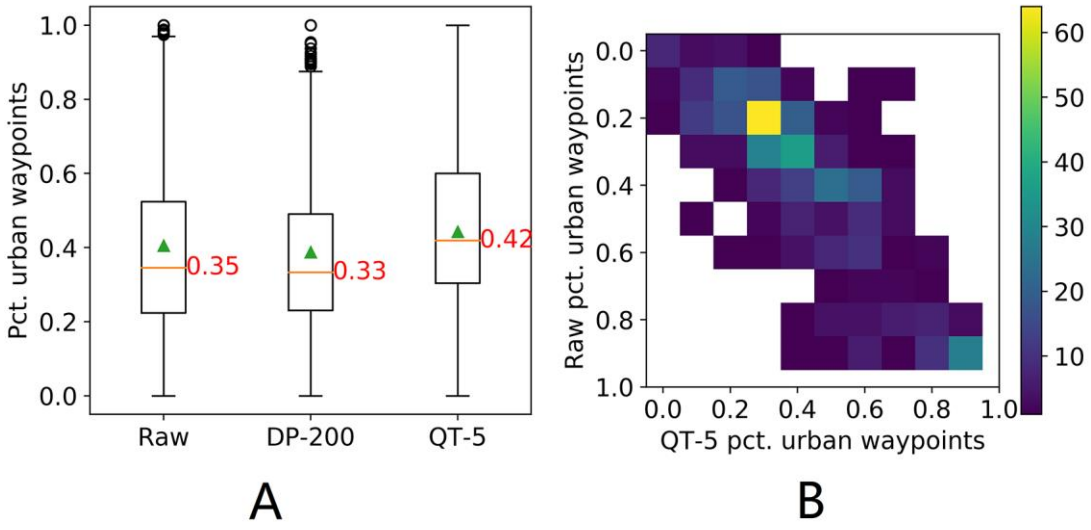


Figure 11 A: Percentages of urban waypoints located in CORINE urban land cover areas of the raw trajectories (N=500), the simplified trajectories by DP-200, and the trajectories simplified by the POI-quadtree with splitting threshold of 5 (QT-5). B: Transition of the percentage of waypoints located in the urban area between the raw trajectories and their corresponding trajectories simplified by QT-5.

4.4 Preservation of pairwise similarities/distances between trajectories

For many common clustering methods such as hierarchical clustering, the same clustering results can still be achieved after data transformation/simplification if the order of distances can still be preserved. Therefore, to explore the distance preservation after trajectory simplification, pairwise distances of any two raw trajectories, and those of simplified trajectories by the proposed POI-quadtree method, were computed after projection. We approached the assessment of distance preservation with three metrics from two perspectives: the geometry perspective and the movement parameter perspective. For the geometry perspective, LCSS (longest common sub-sequence) and DTW (dynamic time warping),

which are commonly employed in existing clustering studies (Kim and Mahmassani 2015, Yuan *et al.* 2017), were chosen as metrics. Both of them are elastic metrics that allow the two trajectories to have different lengths regarding the number of waypoints. For movement parameters, NWED, which is based on symbolizing the sinuosity of movement parameter profiles was employed, and speed was chosen as an example movement parameter. Spearman's rho was used to evaluate the preservation of relative distances between trajectories. A high rho value indicates that the relative distances of the raw trajectories are not changed after simplification, which may lead to little change for many trajectory clustering algorithms.

As the benchmark, the time of calculating 124,750 pairs of LCSS distances, DTW distances, and NWED distances for the raw 500 trajectories was 4348 seconds, 11486 seconds, and 7519 seconds, respectively. The following evaluates how well the pairwise distances between trajectories were preserved after applying the proposed POI-quadtree simplification, while reducing the computing time.

Figure 12 shows the results for the LCSS distances. In general, the relative distances between trajectories remain well preserved after simplification by the POI-quadtree, while the time consumption significantly decreases compared to that of the raw trajectories, due to the reduction of the number of waypoints. For example, with a splitting threshold of 5, the LCSS distance correlation between the pairwise distances of the original trajectory and the pairwise distances of the simplified ones is about 0.75, while the computing time is reduced to just 10% of the original computation time. Correlations tend to decrease with increasing thresholds, which make a trajectory more simplified.

The correlations of DP and TD-TR results are (slightly) higher than the results of the POI-quadtree simplification, depending on the parameters used. This may due to the 200-meter matching distance for the LCSS, and the compression ratios of DP and TD-TR being

lower than with POI-quadtree simplification. With the splitting threshold increasing for the POI-quadtree, the cells in the countryside and suburban areas are more likely to be merged into larger cells than those in the urban area, potentially causing representative waypoints being unmatched.

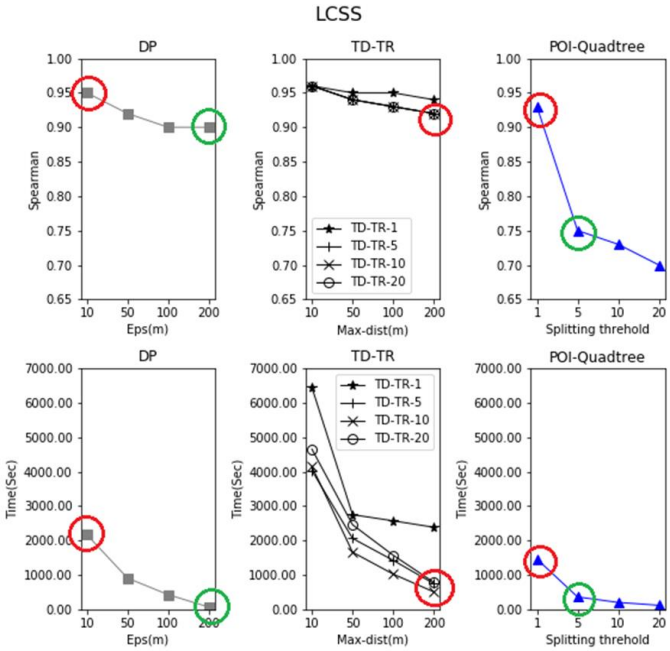


Figure 12 Top row: Correlations between the pairwise LCSS distances of the original trajectories and those simplified by the DP, TD-TR and POI-quadtree. Bottom row: Computing time for corresponding cases in the top row. The maximum required distance for identifying matching waypoints is 200 meters. Circled records with the same color have a similar compression ratio.

Figure 13 shows the results for the DTW distances. Similarly, the relative DTW distances were well kept after simplification by the POI-quadtree, with the computing time being significantly reduced. The distance preservation is even better than for the LCSS distance, and remains at the level of DP and TD-TR, sometimes even better. For example, with a splitting threshold of 5, the DTW distance correlation between the original trajectories and the simplified ones is still about 0.91, while the computation time is reduced to just 15%

of the original computation time. However, DTW is not sensitive to either changing the splitting threshold or changing the layout of POIs, as all correlations remain around 0.90. In addition, the computing time for DTW is higher than for LCSS in most cases (but better than for DP and TD-TR).

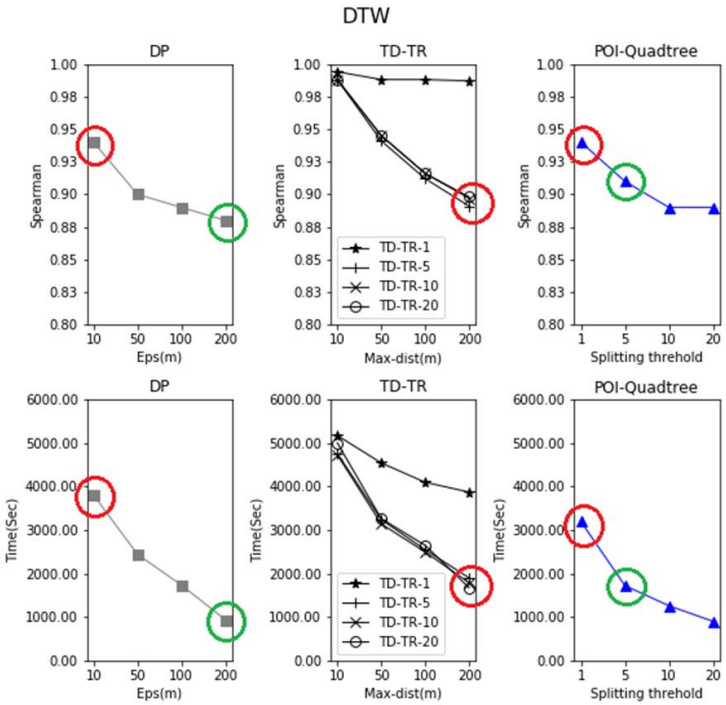


Figure 13 Top row: Correlations between the pairwise DTW distances of the original trajectories and those simplified by the DP, TD-TR, and POI-quadtree. Bottom row: Computing time for corresponding cases in the top row. Circled records with the same color have a similar compression ratio.

Figure 14 shows the results regarding the preservation of movement parameter similarity metrics (i.e., NWED based on speed). In other words, the results demonstrate how well the speed profiles of the original trajectories were kept after simplification. Similarly to the results for the geometric distances (i.e., LCSS and DTW), the correlation of NWED distances between the original trajectories and the simplified ones remained high, with the computing time being significantly reduced. The POI-quadtree based simplification clearly

outperforms DP, as well as most of the TD-TR results. TD-TR only performs better than the POI-quadtree method for very low maximum speed thresholds (max-spd = 1) and maximum distance thresholds (max-dist = 10), respectively. However, with these parameterizations, TD-TR achieves a significantly worse compression ratio than the POI-quadtree method.

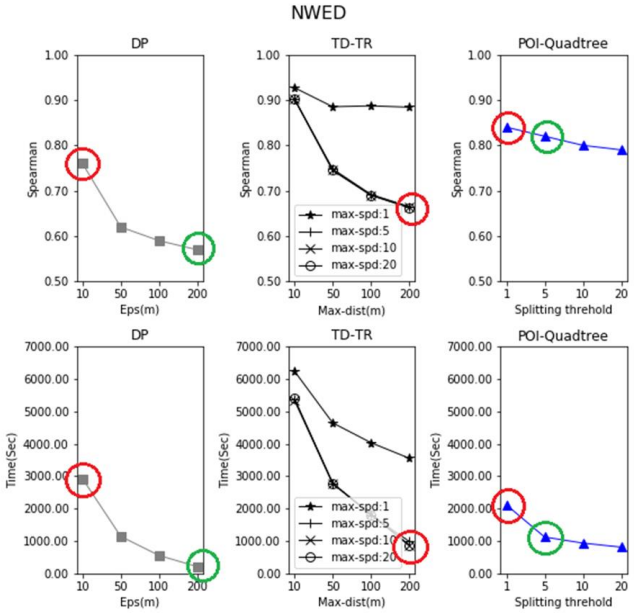


Figure 14 Top row: Correlations between the pairwise NWED distances of the original trajectories and those simplified by the DP, TD-TR and POI-quadtree. Bottom row: Computing time for corresponding cases in the top row. Circled records with the same color have a similar compression ratio.

4.5 Summary of the experimental results

The main findings of the experiments can be summarized as follows:

1) The spatial layout of the built quadtree cells corresponds to the spatial heterogeneity of the POIs, manifesting that the urban areas yield more small cells while the suburban and rural areas obtain more large cells. This suggests that the built POI-quadtree and its resulting spatial subdivision reflects the urban-rural dichotomy of a geographic area.

2) In terms of compression ratio and speed-up of computation, the proposed POI-quadtree method achieves similar results to the DP baseline algorithm, and considerably better results than TD-TR baseline.
3) Regarding simplification errors (using PED, SED and NWED), the POI-quadtree method achieves results similar to those DP and TD-TR when comparable parameter settings are used. The POI-quadtree algorithm has the definite advantage, though, of being able to adapt the simplification error and compression ratio to the underlying urban-rural geographic context, retaining higher spatial resolution and accuracy in urban areas, where it really matters in many applications.
4) The trajectories simplified by the POI-quadtree method can still preserve inter-trajectory distances (i.e., LCSS, DTW and NWED) of the original trajectories. The computing time of the POI-quadtree method can be significantly lower than DP and TD-TR in some cases when similar or better performances are achieved. For the speed-related NWED metric, the results are in most cases considerably better than those of the baseline algorithms.

5. Discussion

This work proposes a quadtree based trajectory simplification method, which makes use of the spatial heterogeneity of POIs as a proxy to model the urban-rural dichotomy, and uses the resulting variable-resolution spatial subdivision to provide adaptive simplifications of trajectories, especially trajectories with large geographic coverage.

The experiments with a real-world vehicle trajectory dataset show that the proposed method can achieve similar results as existing baseline methods (particularly DP and TD-TR) in terms of compression ratio and the overall geometric error introduced by the simplification process. However, in contrast to these existing methods, the proposed POI-quadtree method achieves higher compression outside of cities and lower compression inside cities, as

designed, and should thus be particularly useful in support of transportation-related applications of big mobility data analytics over large areas such as nationwide or international.

The results of the pairwise distance preservation show that introducing POIs as the spatial context for aggregating waypoints can preserve the inter-trajectory distances, while consuming much less computing time, as well as memory due to the shorter lengths of the simplified trajectories. The results of LCSS and DTW based experiments show that DTW is more robust to spatial subsampling. However, DTW has a higher computational cost compared to LCSS.

We also observed the influence of scale in this study. There is a tradeoff between the splitting threshold used and the preservation of the distance metrics, as larger thresholds tend to decrease the preservation accuracy while costing less time. The finding of the scale-effect can help to scale up similarity assessment for much larger trajectory data sets in the context of big data scenarios.

This method could also be used to help visualize large trajectory data sets on web GIS clients. Since waypoints have been indexed by quadtree leaf nodes, the locations of the waypoints can be visualized as a choropleth map of quadtree cells with the color of the rectangles representing the number of raw waypoints per quadtree node. Due to the hierarchical structure of the quadtree, the leaf cells and associated data can easily be aggregated to larger cells, providing efficient interactive operations for the web GIS users to explore the data at different scales. In addition, the method could contribute to trajectory modeling on parallel computing platforms because the representative waypoints are indexed by the POI-quadtree after the simplification. Therefore, the simplified trajectories could be further partitioned over different machines by the spatial index from the POI-quadtree.

POIs inherently have different types. In this study, we used the full POI data set for enriching the waypoints of vehicles. Depending on the target application domain, other datasets may also be employed for the quadtree subdivision, corresponding to other geographic phenomena. For example, to focus on modeling trajectories on highways, certain POI types such as highway intersections, bridges, and ramps might be selected, instead of all POIs. The applications of this method are not limited to vehicle trajectory modeling with land use as the geographic context. Phenomena of physical geography might also be used by this method for modeling other types of moving objects. For example, researchers might want to preserve more trajectory detail near the coast where there are more islands and submerged rocks but preserve less detail in the sea far off the coast while simplifying vessel trajectories. In such a scenario, we could use the location of the islands and rocks for building the quadtree. Similarly, this may help to model animal trajectories in ecology as well. For example, bird migration modeling may preserve more detail of the movement when the birds approach stopover sites but preserve less detail during long-haul displacements (Demšar *et al.* 2015).

The proposed method has several limitations. Quadtrees always provide a grid-based tessellation, which can potentially be sensitive to the spatial distribution of the point features. In some cases, removing or adding a single POI may change the quadtree cells dramatically. The rectangle-shaped tessellation is also a tradeoff for speeding up POI-quadtree building and waypoint assignment but may lead to different sensitivity along different axes. However, it is subject to both the spatial distributions of the POIs and the waypoints. As a method using external geographic information for spatial enrichment, the results of our proposed approach also rely on the quality of the external data set, e.g., the completeness of the OSM POIs in this study. Assigning waypoints to quadtree cells might be sensitive to the positioning uncertainty and errors of the GPS waypoints, which then may lead to error in the trajectory

compression and preservation. The degree of such sensitivity and the corresponding data cleaning strategy should be further investigated.

6. Conclusions and future work

We proposed a novel adaptive trajectory simplification method using a quadtree to allow simplifying segments of a trajectory at different compression ratios according to their underlying geographic context, particularly, the urban-rural configuration. Compared to the existing baseline trajectory simplification methods DP and TD-TR, the proposed method can still achieve similar overall compression ratios and simplification errors. More importantly, the proposed method is able to preserve inter-trajectory distances, while significantly reducing the computing time. The proposed method is particularly suitable for simplifying trajectories with large geographic coverage, e.g., a trajectory of a truck that travels around a whole country or several countries, or a trajectory of a migrating bird.

As future work, we plan to apply the proposed trajectory simplification method on big trajectory datasets, and see whether the spatio-temporal mobility patterns, such as clustering results using the pairwise distance matrix and the median trajectory of the clusters, can still be maintained after simplification. Meanwhile, we are also interested in adopting the proposed method to a distributed computing infrastructure, such as Hadoop and Spark. Further, we will also explore the use of other geographic data such as the road network as inputs when building the quadtree, and investigate its potential applications.

Acknowledgments

The authors would like to thank Vodafone Innovus for providing the trajectory data. The authors also appreciate the comments of three anonymous reviewers which helped improve the paper.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780754.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data and code availability statement

The data and codes that support the findings of this study are available with a DOI at https://doi.org/10.6084/m9.figshare.11708994. The vehicle trajectory data cannot be made publicly available to protect the privacy of research participants. Simulated vehicle trajectories are provided via the link for demonstration purposes.

References

Alvares, L.O., Bogorny, V., Kuijpers, B., de Macedo, J.A.F., Moelans, B., Vaisman, A., Fernandes, J.A., Moelans, B., and Vaisman, A., 2007. A model for enriching trajectories with semantic geographical information. *In*: *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems - GIS '07*. New York, New York, USA: ACM Press, 1.

Andrienko, G., Andrienko, N., and Wrobel, S., 2007. Visual analytics tools for analysis of movement data. *ACM SIGKDD Explorations Newsletter*, 9 (2), 38.

Besse, P.C., Guillouet, B., Loubes, J.-M., and Royer, F., 2016. Review and Perspective for Distance-Based Clustering of Vehicle Trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 17 (11), 3306–3317.

Chen, M., Xu, M., and Franti, P., 2012. A Fast O(N) Multiresolution Polygonal Approximation Algorithm for GPS Trajectory Simplification. *IEEE Transactions on Image Processing*, 21 (5), 2770–2785.

Copernicus, 2019. CLC 2018 [online]. Available from: https://land.copernicus.eu/pan-european/corine-land-cover/clc2018 [Accessed 20 Sep 2019].

Das, R.D. and Winter, S., 2016. Detecting Urban Transport Modes Using a Hybrid Knowledge Driven Framework from GPS Trajectory. *ISPRS International Journal of Geo-Information*, 5 (11), 207.

Demšar, U., Buchin, K., Cagnacci, F., Safi, K., Speckmann, B., Van de Weghe, N., Weiskopf, D., and Weibel, R., 2015. Analysis and visualisation of movement: an interdisciplinary review. *Movement Ecology*, 3 (1), 5.

Dodge, S., 2019. A Data Science Framework for Movement. *Geographical Analysis*, 0, 1–21.

Dodge, S., Laube, P., and Weibel, R., 2012. Movement similarity assessment using symbolic representation of trajectories. *International Journal of Geographical Information Science*, 26 (9), 1563–1588.

Dodge, S., Weibel, R., and Forootan, E., 2009. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33 (6), 419–434.

Douglas, D.H. and Peucker, T.K., 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *Canadian Cartographer*, 10 (2), 112–122.

Fan, J., Fu, C., Stewart, K., and Zhang, L., 2019. Using big GPS trajectory data analytics for vehicle miles traveled estimation. *Transportation Research Part C: Emerging Technologies*, 103 (March), 298–307.

Finkel, R.A. and Bentley, J.L., 1974. Quad Trees A Data Structure for Retrieval on Composite Keys. *Acta Informatica*, 4 (1), 1–9.

Furletti, B., Cintia, P., Renso, C., and Spinsanti, L., 2013. Inferring human activities from GPS tracks. *In*: *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing - UrbComp '13*. New York, New York, USA: ACM Press, 1.

Gibson, J., 1979. The Theory of Affordances. *In*: *The Ecological Approach to Visual Perception*. Boston, MA: Lawrence Erlbaum Associates, Inc., 127–137.

Guidotti, R., Trasarti, R., Nanni, M., Giannotti, F., and Pedreschi, D., 2017. There's a Path for Everyone: A Data-Driven Personal Model Reproducing Mobility Agendas. *In*: *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Tokyo, Japan: IEEE, 303–312.

Hershberger, J. and Snoeyink, J., 1992. *Speeding Up the Douglas-Peucker Line-Simplification Algorithm*. Vancouver, Canada.

Jiang, B. and Brandt, S.A., 2016. A Fractal Perspective on Scale in Geography. *ISPRS*

International Journal of Geo-Information, 5 (6), 95.

Kim, J. and Mahmassani, H.S., 2015. Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories. *Transportation Research Procedia*, 9, 164–184.

Krueger, R., Thom, D., and Ertl, T., 2015. Semantic Enrichment of Movement Behavior with Foursquare–A Visual Analytics Approach. *IEEE Transactions on Visualization and Computer Graphics*, 21 (8), 903–915.

Leichsenring, Y.E. and Baldo, F., 2019. An evaluation of compression algorithms applied to moving object trajectories. *International Journal of Geographical Information Science*, 00 (00), 1–20.

Li, Z., Wang, J., Tan, S., and Xu, Z., 2018. Scale in Geo-information Science: An Overview of Thirty-year Development. *Geomatics and Information Science of Wuhan University*, 43 (12), 2233–2242.

Lin, C.-Y., Hung, C.-C., and Lei, P.-R., 2016. A velocity-preserving trajectory simplification approach. *In*: *2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. Hsinchu, Taiwan: IEEE, 58–65.

Long, C., Wong, R.C.W., and Jagadish, H. V., 2013. Direction-preserving trajectory simplification. *Proceedings of the VLDB Endowment*, 6 (10), 949–960.

Long, Y., Shen, Y., and Jin, X., 2016. Mapping Block-Level Urban Areas for All Chinese Cities. *Annals of the American Association of Geographers*, 106 (1), 96–113.

Meratnia, N. and de By, R.A., 2004. Spatiotemporal Compression Techniques for Moving Point Objects. *In*: *Lecture Notes in Computer Science ·*. 765–782.

Morris, B. and Trivedi, M., 2009. Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. *In*: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, USA: IEEE, 312–319.

Nathan, R., Getz, W.M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., and Smouse, P.E., 2008. A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences*, 105 (49), 19052–19059.

Pelekis, N., Kopanakis, I., Kotsifakos, E.E., Frentzos, E., and Theodoridis, Y., 2011. Clustering uncertain trajectories. *Knowledge and Information Systems*, 28 (1), 117–147.

Purves, R.S., Laube, P., Buchin, M., and Speckmann, B., 2014. Moving beyond the point: An agenda for research in movement analysis with real data. *Computers, Environment and Urban Systems*, 47, 1–4.

Quddus, M. and Washington, S., 2015. Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies*, 55, 328–339.

Ramm, F., 2017. OpenStreetMap Data in Layered GIS Format [online]. *Geofabrik*. Available from: https://www.geofabrik.de/data/geofabrik-osm-gis-standard-0.7.pdf [Accessed 10 Sep 2018].

Richter, K.F., Schmid, F., and Laube, P., 2012. Semantic trajectory compression: Representing urban movement in a nutshell. *Journal of Spatial Information Science*, 4 (2012), 3–30.

Rothermel, K., Schnitzer, S., Lange, R., Dürr, F., and Farrell, T., 2012. Context-aware and quality-aware algorithms for efficient mobile object management. *Pervasive and Mobile Computing*, 8 (1), 131–146.

Samet, H., 2006. *Foundations of Multidimensional and Metric Data Structures*. 1st ed. San Francisco, USA: Morgan Kaufmann Publishers.

Sandu Popa, I., Zeitouni, K., Oria, V., and Kharrat, A., 2015. Spatio-temporal compression of trajectories in road networks. *GeoInformatica*, 19 (1), 117–145.

Shang, S., Xie, K., Zheng, K., Liu, J., and Wen, J.R., 2015. VID Join: Mapping Trajectories to Points of Interest to Support Location-Based Services. *Journal of Computer Science and Technology*, 30 (4), 725–744.

Siła-Nowicka, K., Vandrol, J., Oshan, T., Long, J.A., Fotheringham, A.S., Vandrol, J., Oshan, T., Long, J.A., Demšar, U., and Fotheringham, A.S., 2016. Analysis of human mobility patterns from GPS trajectories and contextual information. *International Journal of Geographical Information Science*, 30 (5), 881–906.

Soleymani, A., Cachat, J., Robinson, K., Dodge, S., Kalueff, A. V, Weibel, R., Jonathan, M., and Allan, V., 2014. Integrating cross-scale analysis in the spatial and temporal domains for classification of behavioral movement. *Journal of Spatial Information Science*, 8 (8), 1–25.

Sun, Z., Zhang, H., Liu, Z., Xu, C., and Wang, L., 2016. Migrating GIS Big Data Computing

from Hadoop to Spark: An Exemplary Study Using Twitter. *In*: *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. San Francisco, CA: IEEE, 351–358.

Toohey, K. and Duckham, M., 2015. Trajectory similarity measures. *SIGSPATIAL Special*, 7 (1), 43–50.

Tuan, Y.-F., 1975. Place: An experiential perspective. *Geographical Review*, 65 (2), 151–165.

Vrotsou, K., Janetzko, H., Navarra, C., Fuchs, G., Spretke, D., Mansmann, F., Andrienko, N., and Andrienko, G., 2015. SimpliFly: A Methodology for Simplification and Thematic Enhancement of Trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 21 (1), 107–121.

Wong, D., 2009. The Modifiable Areal Unit Problem (MAUP). *In*: A. Fotheringham and P. Rogerson, eds. *The SAGE Handbook of Spatial Analysis*. London, UK: SAGE Publications, Ltd, 105–123.

Yuan, G., Sun, P., Zhao, J., Li, D., and Wang, C., 2017. A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review*, 47 (1), 123–144.

Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., and Huang, Y., 2010. T-drive. *In*: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*. New York, New York, USA: ACM Press, 99.

Yuan, J., Zheng, Y., Zhang, C., Xie, X., and Sun, G.-Z., 2010. An Interactive-Voting Based Map Matching Algorithm. *In*: *2010 Eleventh International Conference on Mobile Data Management*. Kansas City, MO: IEEE, 43–52.

Zaharia, M., Franklin, M.J., Ghodsi, A., Gonzalez, J., Shenker, S., Stoica, I., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., and Venkataraman, S., 2016. Apache Spark: a unified engine for big data processing. *Communications of the ACM*, 59 (11), 56–65.

Zhang, D., Ding, M., Yang, D., Liu, Y., Fan, J., and Shen, H.T., 2018. Trajectory simplification: An Experimental Study and Quality Analysis. *Proceedings of the VLDB Endowment*, 11 (9), 934–946.

Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y., 2008. Understanding mobility based on

GPS data. *In*: *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*. New York, USA: ACM Press, 312–321.

Zhuang, C., Yuan, N.J., Song, R., Xie, X., and Ma, Q., 2017. Understanding People Lifestyles: Construction of Urban Movement Knowledge Graph from GPS Trajectory. *In*: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. California: International Joint Conferences on Artificial Intelligence Organization, 3616–3623.

Appendices

Appendix A The pseudo-code of building a quadtree using a set of POIs as input. The algorithm partitions the geographic area by recursively subdividing it into four quadrants or cells. The partition continues if the number of contained POI data points in a cell exceeds a pre-defined splitting threshold.

```
1.  class Quadnode {
2.      size: int //the number of data points within the quadnode's boundary, including its child nodes
3.      bbox: (double, double, double, double) //the min_x, min_y, max_x, max_y of the bounding box of the quadnode
4.      depth: int //the depth of the node in the quadtree; the root has depth 0
5.      northwest, northeast, southwest, southeast: Quadnode* //pointers to its 4 child nodes, if existing; default values NULL
6.      pt_pointers: Point*[]  //a set of data points within the bounding box of the quadnode; only leaf node contains data points
7.      FUNCTION contains(pt) {...} //Return true if a point is within the boundary of the quadnode
8.      ...
9.  }

Algorithm I: Create a quadtree from a set of POIs
10. INPUT
11. POIs = {p1, p2, ..., pN} //POIs as a set of three-item tuples pi=(xi, yi, ti)
12. ST //splitting threshold for partitioning a node into four child nodes
13. MD //the maximal depth of the built quadtree
14. OUTPUT
15. root //the root of the quadtree

16. root ← {size: 0; bbox: (-180,-90,180,90); depth: 0} //the spatial extent of the root is set as the whole world, e.g., in WGS 84
17. FOREACH poi IN POIs
18.     insert_point_to_quadnode (root, poi)
19. ENDFOREACH

20. FUNCTION insert_point_to_quadnode (quadnode*, pt*)
21.     quadnode.size ← quadnode.size + 1
22.     add pt to quadnode.pt_pointers
        //create four child nodes for the current quadnode if these three conditions meet: 1) the number of data points in the
        quadnode exceeds the splitting threshold, 2) it does not have any child nodes yet, 3) it does not reach the maximal depth
23.     IF quadnode.size >= ST && quadnode.northeast == NULL && quadnode.depth < MD
            //function create_child_quadnode returns a quadnode, setting its size, bbox and depth accordingly
24.         quadnode.northwest ← create_child_quadnode (quadnode, 1)
25.         quadnode.northeast ← create_child_quadnode (quadnode, 2)
26.         quadnode.southwest ← create_child_quadnode (quadnode, 3)
27.         quadnode.southeast ← create_child_quadnode (quadnode 4)
28.     ENDIF
29.     IF quadnode.northwest != NULL
30.         FOREACH pt_parent IN quadnode.pt_pointers // move the data points of the current quadnode to its child nodes
31.             IF quadnode.northwest.contains (pt_parent)
32.                 insert_point_to_quadnode (quadnode.northwest, pt_parent)
33.             ELSEIF quadnode.northeast.contains (pt_parent)
34.                 insert_point_to_quadnode (quadnode.northeast, pt_parent)
35.             ELSEIF quadnode.southwest.contains (pt_parent)
36.                 insert_point_to_quadnode (quadnode.southwest, pt_parent)
37.             ELSEIF quadnode.southeast.contains (pt_parent)
38.                 insert_point_to_quadnode (quadnode.southeast, pt_parent)
39.             ENDIF
40.         ENDFOREACH
41.         quadnode.pt_points ← {}
42.     ENDIF
43. ENDFUNCTION

44. RETURN root
```

Two adjustments are made in Line 13 for the maximal depth of the quadtree to control the minimal size of the leaf cell and in Line 16 for the spatial extent of the root as the extent of the world.

Line 23 recursively checks whether the number of contained POI data points exceeds the ST and whether it has not reached the MD. If yes, the associated geographic area is partitioned

into four quadrants, or cells by creating four child nodes (Lines 24-27), and the contained POI points are distributed to the corresponding child nodes (Lines 30-41). The output of the algorithm is the built POI-quadtree (Line 44).

Appendix B The pseudo-code of the enrich-and-aggregate procedure for simplifying a trajectory by a POI-quadtree

Algorithm II: Simplify a trajectory by a POI-Quadtree

1. **INPUT**
2. *traj* = {p_1, p_2, …, p_N} //original waypoints as a sequence of tuples p_i=(x_i, y_i, t_i)
3. *root_QT* //the root of a POI-quadtree; this can be an output of Algorithm I "create a quadtree from a set of POIs"
4. **OUTPUT**
5. *traj'* //a sequence of aggregated waypoints (after simplification)

6. *traj'* ← {}
7. *current_leaf* ← NULL
8. *current_segment* ← {}

   //for each original waypoint *p*, check whether its previous sibling waypoint shares the same leaf node with it. If yes, add it to the *current_segment*. If not, aggregate all waypoints in *current_segment* into a new waypoint, add the new waypoint to *traj'*, and replace the *current_segment* with a new set {*p*}.
9. **FOREACH** *p* **IN** *traj*
10. **IF** *current_leaf* == NULL
        //function p_in_QT_leaf returns the POI-quadtree leaf node that the waypoint *p* is located in
11. *current_leaf* = **p_in_QT_leaf** (*root_QT*, *p*)
12. add *p* to *current_segment*
13. **ELSEIF** *current_leaf* == **p_in_QT_leaf** (*root_QT*, *p*)
14. add *p* to *current_segment*
15. **ELSEIF** *current_leaf* != **p_in_QT_leaf** (*root_QT*, *p*)
        //function aggregate_WPs aggregates several original waypoints of the *current_segment* into a representative one
16. *p'* ← **aggregate_WPs** (*current_segment*)
17. add *p'* to *traj'*
18. *current_segment* ← {*p*}
19. *current_leaf* = **p_in_QT_leaf** (*root_QT*, *p*)
20. **ENDIF**
21. **ENDFOREACH**

    //aggregate all waypoints in *current_segment* to be the last waypoint of the simplified trajectory
22. *p'* ← **aggregate_WPs** (*current_segment*)
23. add *p'* to *traj'*

24. **RETURN** *traj'*

Algorithm 2 takes a raw trajectory and the root of the built quadtree from Algorithm 1, Appendix A as inputs, and returns a simplified trajectory.

Lines 9-21 iteratively check whether a waypoint is located in the same quadtree leaf node with its previous sibling waypoints, and aggregates them into a representative waypoint if a new leaf node appears. The representative waypoint (computed via the function on Lines 16 and 22) might be a new point, taking the geometric centroid of the original waypoints as its location and the midpoint of time of the first and last original waypoints in the leaf node as its timestamp.

Note that to improve the association of the original waypoints to the leaf node in the quadtree (i.e., p_in_QT_leaf function on Lines 11, 13, 15, and 19), the nodes of the quadtree can be coded by the Morton code (i.e., z-order curve, Morton 1966) or other spatial filling curves. Additionally, time constraints may also be added in Lines 9-21, e.g., to prevent waypoints from being aggregated together if the time span between them exceeds some threshold.